



HAL
open science

Decomposition methods for the beam-layout optimization problem

Camille Lescuyer, Christian Artigues, Jean-Thomas Camino, Cédric Pralet

► **To cite this version:**

Camille Lescuyer, Christian Artigues, Jean-Thomas Camino, Cédric Pralet. Decomposition methods for the beam-layout optimization problem. 2025. hal-04901891v2

HAL Id: hal-04901891

<https://hal.science/hal-04901891v2>

Preprint submitted on 23 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Decomposition methods for the beam-layout optimization problem

Camille Lescuyer^a, Christian Artigues^c, Jean-Thomas Camino^a, Cédric Pralet^b

^a*Airbus Defence and Space, 31 Rue des Cosmonautes, Toulouse, 31400, France*

^b*ONERA, 2 Avenue Marc Pelegrin, Toulouse, 31400, France*

^c*LAAS-CNRS, 7 Avenue du Colonel Roche, Toulouse, 31400, France*

Abstract

In this article, we study an optimization problem related to the design of a telecommunication satellite, namely the beam-layout optimization problem in which the goal is to define the beams emitted by the antennas of a geostationary satellite to cover regions on Earth. Four key features of the application tackled are that (1) the regions to cover are defined as polygons, (2) there is a very large number of candidate beams, (3) the selected beams need to be colored (actually, allocated to an antenna reflector) and two close beams cannot have the same color, and (4) the candidate beams have heterogeneous sizes and the sizes of the selected beams must be minimized for efficiency reasons. We provide a complexity analysis, showing that the problem is NP-hard. To solve this challenging problem, we introduce two decomposition methods based on column generation and logic-based Benders decomposition, to go beyond the existing heuristic approaches. The experimental results show that these decomposition methods provide high-quality solutions within limited computational times, and that the logic-based Benders decomposition approach finds the optimal solution for many instances.

Keywords: Combinatorial optimization, OR in telecommunications, Column Generation, Logic-Based Benders Decomposition

1. Introduction

The design of the payload of a telecommunication satellite on a geostationary orbit is driven by the mission it must provide: the characteristics of the areas towards which data must be emitted (geographical position, size,

shape) and the demand of the users on the ground (distribution, type). From the set of requirements of a satellite’s operators or service provider, satellite system manufacturers must develop a whole comprehensive solution specifying the ground segment (gateways and control centers), the user segment (satellite terminals), and the space segment (the telecommunication satellite itself). Designing these three segments altogether is impractical due to the numerous decisions and computationally intensive simulations required. This is why the standard approach breaks down the problem into simpler sub-problems. In this context, the *beam-layout optimization problem* illustrated in Figure 1 is one of the first sub-problems to be solved. It consists in defining the positions, sizes, and shapes of a set of beams emitted by the antenna reflectors (or dishes) of the telecommunication satellite to cover the demand of the end-users on the ground, while minimizing the sizes of the beams used for efficiency reasons. In many cases, satellite manufacturers exploit multi-beam antennas where each antenna reflector (usually three or four reflectors) can emit several highly focused beams providing high-capacity and efficient coverage. But when using a standard technology such as *Single-Feed-Per-Beam* (SFPB) antennas, one constraint is that two beams emitted by the same reflector cannot be too close to each other.

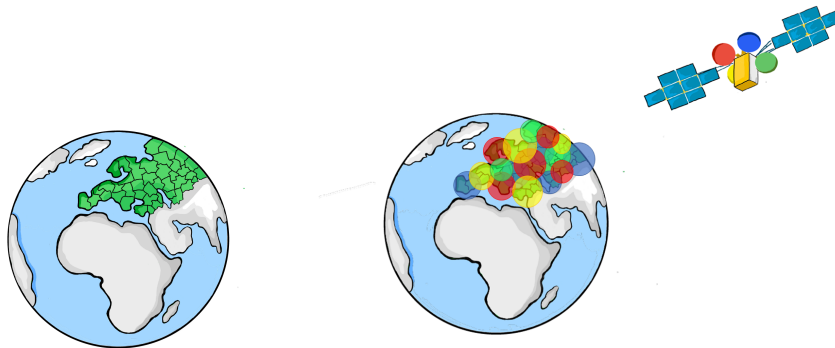


Figure 1: Beam-layout for a multi-beam telecommunication satellite: (left) set of polygons on the Earth surface, in green; (right) beams emitted by four antenna reflectors to cover the polygons, with one color per reflector and several beams for each reflector

Globally, defining the beams emitted by the satellite and choosing the antenna reflector allocated to each beam is a challenging optimization problem. This problem encompasses on one hand a geometric covering problem since a first expected goal is to define a set of beams of minimum sizes covering a set of polygons, and on the other hand a coloring problem since a second

objective is to associate a color (a reflector) to each beam so that two close beams have different colors. Previous work on the considered beam-layout optimization problem consists of an exact method exploiting a basic Integer Linear Programming (ILP) model and two incomplete methods based on matheuristics and metaheuristics [16]. However, the existing basic ILP approach does not scale well and the existing heuristic methods do not bring any guarantee on the quality of the solutions found. This article aims to explore the use of two decomposition methods, namely column generation and logic-based Benders decomposition, with the ambition of producing high-quality solutions and, if possible, proving optimal solutions.

The remainder of the article is organized as follows. Section 2 describes related works and our main contributions. Section 3 formalizes our beam-layout optimization problem. Section 4 provides a complexity analysis showing that the problem addressed is NP-hard. Section 5 gives an integer linear programming model. Sections 6 and 7 detail the column generation method and the logic-based Benders decomposition method, respectively. Section 8 provides experimental results on representative instances with regards to the industrial needs. Section 9 concludes and gives perspectives.

2. Related Works

The problem we consider can be related to both generic works on covering and coloring problems in the Euclidean space and works dedicated to beam-layout optimization for telecommunication satellites.

Covering and coloring problems in the Euclidean space. On the coloring side, the beam-layout optimization problem is related to the *Unit Disk coloring problem* that has applications in channel allocation [12]. This problem consists in coloring, with a given number of colors, the intersection graph defined by unit disks in the Euclidean space [6]. In our case, the disks correspond to the beams and the colors correspond to the reflectors. More details are provided in Section 4 dedicated to the complexity analysis.

On the covering side, the beam-layout optimization problem is related to the *Unit Disk cover* problem that consists in covering a set of points P in a 2D plane with m unit disks with free centers. It is also related to the *Discrete Unit Disk cover* problem, where a set of candidate disks D is introduced and the goal is to select of subset of unit disks $D' \subseteq D$ of minimal size to cover the points in P . Using a geometric transformation,

this problem can be reformulated as a set covering problem for which greedy methods can be used [7]. Recent works combine unit disk covering and graph coloring, requiring overlapping disks to have distinct colors. Space partitioning methods with exhaustive search were proposed to address this constraint [19]. Beyond points and unit disks, some authors have studied covering polygons with a union of disks having fixed centers and flexible radii, aiming to minimize the total disk surface area [2]. They proposed two branch-and-bound algorithms optimizing a similar objective function to ours, but these cannot be directly reused as we require each polygon to be *entirely* covered by at least one disk.

Beam-layout optimization. In the literature, several works are dedicated to beam-layout optimization, but they differ on the features of the beams, antennas, and end-user demands as well as on the objective functions optimized and the methods exploited. In [1], the authors demonstrate the efficiency of non-regular multi-beam payloads for addressing non-uniform traffic demands. Evenly loaded polygonal regions are built using Voronoi diagrams according to the distribution of the traffic load. The beam sizes based on the polygons' shape and surface. However, antenna configuration constraints are not addressed. The introduction of antenna configuration constraints, or mapping the different beams to the available reflectors, was first discussed in [15]. They maximize the minimum inter-beam distances which implies the largest possible feed sizes. A joint optimization of non-uniform beam sizes (two sizes considered) and bandwidth allocation enhanced capacity and resource efficiency compared to uniform beam sizes. In [4], the authors introduced a randomized multi-start approach for irregular beam-layouts, allowing beam centers and widths to take discrete values. Beam-to-reflector allocation, modeled as a graph coloring problem, was solved using a dynamic first-fit algorithm and simulated annealing for unsatisfactory cases. An ILP formulation with continuous beam positions and sizes was also proposed in [5], achieving good results on small instances but encountering numerical issues for larger ones. In [9], hybrid methods combining heuristics, ILP, and constraint programming were developed. One heuristic selects a user for initial beam placement, expanding the beam until capacity or size limits are reached. Another employs weighted kMeans clustering to group users, with a custom heuristic for adjustments. Frequency assignment uses a separate graph coloring problem based on a four-color reuse scheme. However, these methods fail to fully meet demand. All the previously mentioned works were

carried out for a traffic modeled as points on Earth with specific bit-per-second demands. Few studies address beam placement for polygon-defined regions while considering antenna constraints and beam-to-reflector allocation. It is therefore of interest to the space industry to develop algorithms to automatically solve this problem. Indeed, in this paper, we concentrate on broadcasting satellites, placed in geostationary orbit to transmit television signals to ground-based receivers. These satellites revolutionized television by enabling the distribution of TV signals over vast distances, beyond what traditional terrestrial transmitters could cover. Usually, broadcasting satellites use a unique global beam covering a whole country or even continent to broadcast the same channels on the whole area of interest. But some companies prefer deploying satellites producing “linguistic beams” or “spot beams” due to the market segmentation. These multiple and small beams take advantage of the satellite’s ability to broadcast different signals to small geographic areas and optimize content delivery based on the needs of the audience. Patent [13] presents several beam-layouts for television broadcasting on several regions of interest on the United-States. They generate non-uniform beams (3 sizes allowed) based on population density within specific polygons. But the paper only treats one use case and the method isn’t proved to work on a set of polygons. In [16], we proposed two heuristic methods to solve the exact same problem of this paper. The merge-and split heuristic iteratively updates a set of beams with local mechanisms of merge and split to find an feasible beam-to-reflector allocation. The second method is a matheuristic: an Integer Linear Programming model (ILP) is solved given an evolving pool of beams, which is build-up at each iteration as long as the solution does not cover all polygons.

Contributions. The heuristics have shown good results to provide solutions in a reasonable computational time and in [16]. A comparison of the solutions of the two heuristic methods one with each other is provided, but it is frustrating not to know if these solutions are close to the optimal ones. Finding solutions using exact methods within reasonable time can validate the heuristic solutions by comparing their quality to exact results. While exact methods may require longer computation times, evaluating whether this extra effort yields significantly higher-quality solutions is worthwhile.

The combinatorial complexity of the problem, with an exponential number of potential beams, makes it a large-scale optimization challenge. For instance, 20 polygons yield over 1048576 subsets of polygons possible, and

as much potential beams. Decomposition approaches, such as Column Generation to restrict variables or Logic-Based Benders Decomposition to limit constraints, are promising strategies.

In the literature, few studies address beam placement for polygon-defined regions with antenna constraints and beam-to-reflector allocation. The heuristics in [16] do not provide an optimality gap, emphasizing the need for exact methods to assess and validate heuristic performance.

3. Problem Formalization

3.1. Polygons

The area of interest is divided into a set of polygons P , each one specified by a list of points in a 2-dimensional space. Each point in this 2-dimensional space describes the angular position of a target on Earth from the point of view of the geostationary satellite.

3.2. Beam definition

Each beam serves an area on Earth where the density of signal is sufficiently high for the end-users to receive data. Also, for the telecommunication mission addressed, each polygon $p \in P$ must be fully covered by at least one beam, *i.e.* it is not sufficient to cover p using a union of beams. In the 2-dimensional space considered, a beam is modeled as a disk defined by a center and a radius s_b . A beam is limited in size: there exists a minimum beam radius $s_{MIN} > 0$ and a maximum radius s_{MAX} due to respectively the smallest and largest manufacturable size of the feeds emitting the beams (physically speaking, a bound on the beam angle in degrees). For the industrial application tackled, antenna specialists specify an upper value $s_{MAX} = 0.75^\circ$. Reducing the value of s_{MAX} allows us to reduce the set of possible beams while preventing the selection of large beams, but care must be taken not to make the problem infeasible. For this, we exploit the maximum beam radius s_{MAX} obtained in the solutions produced by heuristic methods [16].

3.3. Beam database

A beam produced by a geostationary satellite may be centered on any point and may have any size in $[s_{MIN}, s_{MAX}]$. However, in practice, small size beams are preferred so as to optimize the quality of service. This is why we assume that the best beam covering a subset of polygons $P' \subseteq P$ is defined from the smallest circle enclosing the points that define the

polygons in P' . We compute such a smallest enclosing in polynomial time using Welzl's algorithm [23], and if needed we enlarge the radius of this circle to reach value s_{MIN} . All the beams we consider are defined this way and the beam b obtained from the smallest enclosing circle of a subset of polygons $P' \subseteq P$ is referred to as $b = W(P', s_{MIN})$, or more simply $b = W(P')$ since s_{MIN} is a constant input data. This assumption gives us a *finite* set of candidate beams $\mathcal{B} = \{b = W(P') \mid P' \subseteq P, P' \neq \emptyset, s_b \leq s_{MAX}\}$ defined from the coverage of subsets of polygons in P . One difficulty however is that enumerating the $2^{|P|} - 1$ non-empty subsets of polygons in P leads to a combinatorial explosion. But unlike our previous works [16], we exploit a key property according to which the smallest enclosing circle of a set of points Q can be determined by at most three points in Q that lie on the boundary of the circle. The set of candidate beams \mathcal{B} can then be obtained by enumerating only the beams defined by subsets of 3, 2 and 1 polygon(s), that is $\mathcal{B} = \{b = W(P') \mid P' \subseteq P, s_b \leq s_{MAX}, |P'| \in \{1, 2, 3\}\}$. The number of candidate beams is therefore at most cubic in the number of polygons in P .

Table 1 gives the number of beams obtained for several instances, including details concerning the time required to generate these beams and the size of the subsets of beams \mathcal{B}_{2P} and \mathcal{B}_{3P} defined by 2 and 3 border polygons respectively. For example, a beam $b = W(\{p_1, p_2, p_3\})$ defined from 3 distinct polygons p_1, p_2, p_3 belongs to \mathcal{B}_{3P} if and only if $b \neq W(\{p_1, p_2, p_3\} \setminus \{p_i\})$ for every $i \in \{1, 2, 3\}$, that is the 3 polygons are actually needed to defined b . For the instances in Table 1, we can observe that the size of \mathcal{B}_{3P} is lower than the size of \mathcal{B}_{2P} . To get a quadratic number of candidate beams, an alternative is to restrict the enumeration to subsets of 2 and 1 polygon(s).

3.4. Beam-to-reflector allocation

The number of antenna reflectors available is denoted by N_R , and the set of reflectors available is $\mathcal{R} = \{1, \dots, N_R\}$. With the Single-Feed Per Beam (SFPB) antenna technology, the beams emitted through a reflector r are produced by elementary devices called feeds, that are distributed in a cluster placed just under r . As illustrated in Fig. 2, the sizes and positions of the beams emitted through a reflector then depend on the sizes and positions of their feeds in the cluster. The distribution of the beams on the different reflectors needs to take into account design constraints because if two large beams that are close to each other are assigned to the same reflector, there may not be enough place for the two corresponding feeds under the reflector.

Instance	$ P $	s_{MAX} ($^{\circ}$)	$ \mathcal{B} $	$ \mathcal{B}_{2P} $	$ \mathcal{B}_{3P} $	Time (s)
France	22	0.4	520	369	129	162.91
GuyaneSur.	23	0.5	677	484	170	2.24
Italy	30	0.3	862	710	122	3.06
Venezuela	34	0.6	1347	942	371	5.64
Morocco	39	0.5	1706	1408	259	6.41
UK	40	0.3	3946	3162	744	27.34
Spain	44	0.4	5046	3622	1380	609.83
BlackSea	51	0.5	6221	4750	1420	54.14
EU51	51	0.5	8322	6003	2268	177.03
EU52	52	0.5	4628	3920	656	1171.73
Egypt	90	0.5	30694	21778	8826	517.44
WestEU	92	0.5	16800	12539	4169	195.93
EU102	102	0.7	25707	4275	21325	18825.74
EU136	136	0.55	11655	3299	8220	9738.3

Table 1: Computation of the beam database for different instances

In the literature, these restrictions are modeled by beam separation constraint imposed over two beams assigned to the same reflector [4]. If $d(b_1, b_2)$ denotes the Euclidean distance between the centers of two beams b_1 and b_2 , this separation constraint is expressed as $d(b_1, b_2) > \kappa(s_{b_1} + s_{b_2})$ where κ is a parameter fixed by antenna specialists. This amounts to saying that the beams enlarged by a factor κ must not overlap.

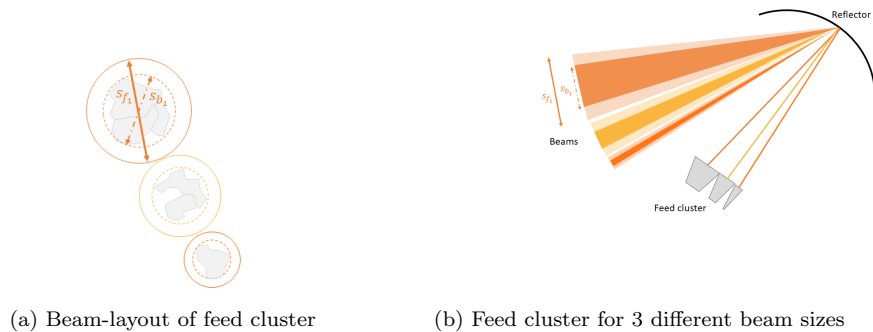


Figure 2: Feed cluster and associated beams

3.5. Objective functions

The beam-layout and the mapping of beams to reflectors do not suffice to precisely evaluate the performances of our system, due to missing features

related to frequency assignment, power distribution, end-users terminals, etc. These aspects are detailed later in the payload design process but we know that smaller beams lead to stronger and more reliable signals for the end-users. This is why our main objective at this design step is to minimize the sizes of the selected beams. Considering $\tilde{\mathcal{B}}$ the set of selected beams, different objective functions can be considered:

- minimize $SRS = \sum_{b \in \tilde{\mathcal{B}}} s_b^2$ (minimize the Squared Radius Sum);
- minimize $MSRS = \frac{1}{|\tilde{\mathcal{B}}|} \sum_{b \in \tilde{\mathcal{B}}} s_b^2$ (minimize the Mean SRS over the beams);
- minimize $MaxS = \max_{b \in \tilde{\mathcal{B}}} s_b$ (minimize the worst beam size);
- maximize $|\tilde{\mathcal{B}}|$ (maximize the total number of beams selected).

In this article, we choose to optimize the **SRS** criterion because (1) it is easier to handle than the MSRS objective that involves term $\frac{1}{|\tilde{\mathcal{B}}|}$ adding a non-linearity (since the number of beams to select is not known beforehand); (2) it has the advantage of impacting the whole set of beams of the solution compared to the maximum radius criteria which only optimizes the worst beam, (3) it is better than the $|\tilde{\mathcal{B}}|$ that may produce solutions involving a few very large beams in exchange for many small ones. Nevertheless, even if we optimize the SRS objective, we take care of evaluating the other criteria on the solutions found.

3.6. Beam-layout optimization problem

Our problem is defined as a tuple $(P, s_{MIN}, s_{MAX}, N_R, \kappa)$ specifying a set of polygons P , a minimum beam radius s_{MIN} , a maximum beam radius s_{MAX} , a number of reflectors N_R , and a beam separation factor κ . From this definition, the set of candidate beams is $\mathcal{B} = \{b = W(P') \mid P' \subseteq P, |P'| \in \{1, 2, 3\}, s_b \leq s_{MAX}\}$ and the set of incompatible beams that cannot be allocated to the same reflector is $I = \{\{b_1, b_2\} \mid b_1, b_2 \in \mathcal{B}, d(b_1, b_2) \leq \kappa(s_{b_1} + s_{b_2})\}$. A solution to this problem:

- selects a subset of beams $\tilde{\mathcal{B}} \subseteq \mathcal{B}$ so that each polygon $p \in P$ is fully included in a beam $b \in \tilde{\mathcal{B}}$ (*i.e.*, there exists $b \in \tilde{\mathcal{B}}$ such that $p \subseteq b$);
- assigns a reflector $r_b \in \{1, \dots, N_R\}$ to each beam $b \in \tilde{\mathcal{B}}$ so that two incompatible beams are allocated to different reflectors (*i.e.*, $r_{b_1} \neq r_{b_2}$ for every pair of incompatible beams $\{b_1, b_2\} \in I$).

A solution is optimal when its SRS ($\sum_{b \in \tilde{\mathcal{B}}} s_b^2$) is minimal among the solutions.

From a graph coloring perspective, if we construct graph $\mathcal{G} = (\mathcal{B}, I)$, the set of beams $\tilde{\mathcal{B}}$ can be emitted by the satellite while satisfying the beam separation constraints if and only if the subgraph of G induced by $\tilde{\mathcal{B}}$ is colorable using at most N_R colors. In the instances we tackle, the number of reflectors available is $N_R = 4$ but the methods we define are applicable to any value of N_R . Also, we use $\kappa = \sqrt{3}$ which was proved to be a good value for multi-beam antennas involving four reflectors

In the following, $P_b \subseteq P$ denotes the subset of polygons covered by a beam $b \in \mathcal{B}$. Conversely, $\mathcal{B}_p \subseteq \mathcal{B}$ refers to the subset of beams which entirely cover a polygon $p \in \mathcal{P}$, and $\mathcal{B}_{\{p_1, \dots, p_n\}} = \mathcal{B}_{p_1} \cap \dots \cap \mathcal{B}_{p_n}$ refers to the set of beams that cover all polygons in $\{p_1, \dots, p_n\}$.

4. Complexity Analysis

In the beam-layout optimization problem, there are two sources of complexity: the covering part and the coloring part. This section provides new complexity results showing that the beam-layout design problem we tackle is NP-hard due to its relationship with two existing problems known as the unit disk covering problem and the unit disk graph coloring problem. To avoid considerations related to the manipulation of real numbers, we prove the result for beam-layout optimization problems $(P, s_{MIN}, s_{MAX}, N_R, \kappa)$ whose basic inputs are integers.

4.1. Complexity of the covering part

To analyze the complexity of the covering part, we use the *unit disk cover problem* which can be stated as: given a set of points Q and a set of m unit disks in the plane (or disks of fixed-radius), is it possible to cover all the points in Q using the m unit disks, given that the center of each disk can be freely chosen. This problem was proven NP-complete in the case where the input data (point coordinates and disk radius) take integer values [8].

Let us transform an instance of this problem into an instance of our beam-layout optimization problem. We consider the set of polygons $P = Q$ (polygons reduced to elementary points) together with integer input data $N_R = m$, $s_{MIN} = s_{MAX} = 1$, and $\kappa = \max\{d(q_1, q_2)^2 \mid q_1, q_2 \in Q\}$. If a solution σ to the beam-layout problem with these settings exists, at most m beams are selected in this solution because it is not possible to allocate two beams to the same reflector due to the large value chosen for κ . As all the

selected beams have a radius equal to 1, solution σ gives us a strategy to cover the points in P using no more than m unit disks. On the other side, let us assume that the points in P can be covered using no more than m unit disks D_1, \dots, D_m . In this case, each unit disk D_k covers a set of points $Q_{D_k} \subseteq Q$ for which the beam-layout problem necessarily contains a candidate beam $\beta_k = W(Q_k)$. In this case, the selection of beams $W(Q_{D_1}), \dots, W(Q_{D_m})$ is a solution to the beam-layout problem. As a result, the points in Q can be covered using m unit disks if and only if the beam-layout problem built has a solution. This proves that our problem is NP-hard even in the case where there is no real choice concerning the allocation of beams to reflectors.

4.2. Complexity of the coloring part

We now consider the *unit disk graph coloring problem* [6]. Basically, a Unit Disk graph is an intersection graph obtained from a set of equal-sized disks D in the plane, that is a graph containing one vertex per disk in D and one edge between two vertices whose associated disks overlap. See Fig. 3a for an illustration. Unit disk graph coloring is proven NP-complete for any chromatic number $k > 3$ [11]. Moreover, for each unit disk graph \mathcal{G} , it is possible to compute in polynomial time an equivalent *proximity graph* defined from a set of points Q having integer coordinates and an integer separation distance d_0 . This proximity graph contains one vertex per point in Q and one edge between two vertices q_1, q_2 whose Euclidian distance $d(q_1, q_2)$ satisfies $d(q_1, q_2) \leq d_0$ [11].

The coloring problem for such a proximity graph using at most k colors can be transformed in polynomial time into our beam-layout optimization problem. The transformation is illustrated in Fig.3b. The main idea is to introduce, for each point $q = (x_q, y_q) \in Q$, one polygon defined as the square losange L_q whose points have coordinates $(2x_q - 1, 2y_q)$, $(2x_q, 2y_q + 1)$, $(2x_q + 1, 2y_q)$, $(2x_q, 2y_q - 1)$, and to consider $N_R = k$ reflectors together with parameters $s_{MIN} = s_{MAX} = 1$, $\kappa = d_0$. With these input data, it can be shown that the polygons in $P = \{L_q \mid q \in Q\}$ never strictly overlap and the set of candidate beams is $\mathcal{B} = \{\beta_q \mid q \in Q\}$ where each beam β_q has center $(2x_q, 2y_q)$ and radius $s_{\beta_q} = 1$. Then, determining whether it is possible to allocate a reflector to each beam under the beam separation constraints is equivalent to determining whether the initial unit-disk graph can be colored using at most k colors. Indeed, two distinct beams β_{q_1}, β_{q_2} in \mathcal{B} of centers $(2x_{q_1}, 2y_{q_1})$ and $(2x_{q_2}, 2y_{q_2})$ can be allocated to the same reflector if and only if the distance between theirs centers is greater than $\kappa \cdot (1 + 1)$, which is

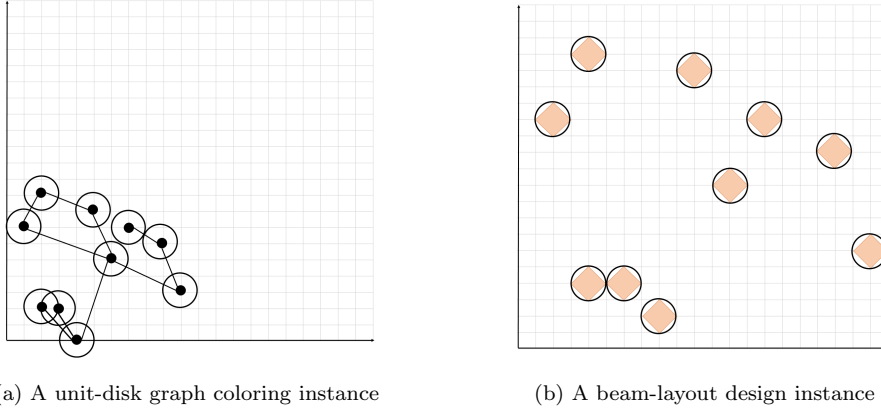


Figure 3: From unit-disk graph coloring to beam-layout design

equivalent to condition $d(q_1, q_2) > d_0$. This proves that our problem is NP-hard, even when the candidate beams cover a unique region.

5. Integer Linear Programming Model

We now recall the integer linear programming model developed for the beam-layout optimization problem [16]. This model is given in Equations (1a-1e), where the decisions are represented by variables $x_{b,r} \in \{0, 1\}$ taking value 1 if and only if beam b is selected and allocated to reflector r . Objective function (1a) consists in minimizing the squared radius sum of the selected beams (SRS objective). Constraints (1b) ensure that each polygon is fully covered by at least one beam. Constraints (1c) ensure that each beam is assigned to at most one reflector. Finally, Constraints (1d) enforce the separation constraints for the beams allocated to the same reflector. Aggregated versions of these separation constraints could be expressed by imposing $\sum_{b \in C} x_{b,r} \leq 1$ for each reflector r and each maximum clique of incompatible beams C .

Instance	P	$s_{MAX}(\circ)$	\mathcal{B}	MaxS	MSRS	SRS	Time (s)	
							Model creation	Solving
France	22	0.4	9	0.372	0.0514	0.462	5.51	2.17
GuySur	23	0.5	6	0.4789	0.08862	0.5317	3.3	6.5
Italy	30	0.3	12	0.1940	0.01643	0.1971	2.6	5.5
Venezuela	34	0.6	7	0.5883	0.16846	1.1794	12.5	38.9
Morocco	39	0.5	9	0.3736	0.0438	0.5694	12.9	19.2
UK	40	0.3	12	0.2290	0.01708	0.2049	47.5	148.1
Spain	44	0.4	9	0.3993	0.04518	0.4067	120.9	418.1
EU51	51	0.5	7	0.4631	0.0815	0.571	291.3	1080.1
EU52	52	0.5	17	0.4738	0.0370	0.629	132.7	42.2
BlackSea	51	0.5	7	0.484	0.0771	0.539	133.3	468.4

Table 2: Exact ILP results obtained using CPLEX12.10 (runs made on a server with 96 cores of an Intel(R) Xeon(R) Gold 5318Y CPU @2.10GHz processor and 62GB of RAM)

$$M_{EXACT} = \text{minimize} \quad \sum_{b \in \mathcal{B}, r \in \mathcal{R}} x_{b,r} \cdot s_b^2 \quad (1a)$$

$$\text{subject to} \quad \sum_{b \in \mathcal{B}_p, r \in \mathcal{R}} x_{b,r} \geq 1, \quad \forall p \in P, \quad (1b)$$

$$\sum_{r \in \mathcal{R}} x_{b,r} \leq 1, \quad \forall b \in \mathcal{B}, \quad (1c)$$

$$x_{b_1,r} + x_{b_2,r} \leq 1, \quad \forall \{b_1, b_2\} \in I, \forall r \in \mathcal{R}, \quad (1d)$$

$$x_{b,r} \in \{0, 1\}, \quad \forall b \in \mathcal{B}, \forall r \in \mathcal{R} \quad (1e)$$

Solving optimally the previous model becomes challenging, if not impossible, for large instances. Table 2 illustrates computation times using CPLEX12.10 for small and medium instances. We distinguish “Model creation time” (importing data, computing set I , and creating the model) from the “Solving time” (reading and solving the model). Model creation is notably time-intensive due to the high number of variables and constraints, highlighting the need for decomposition methods. Larger instances are excluded from the table, as their model creation alone exceeded 2000 seconds, making them impractical to handle.

6. Column Generation

One drawback of the model presented in Section 5 is that it involves a lot of symmetries: for each solution produced, there are $N_R!$ equivalent solutions

corresponding to the possible permutations of the N_R reflectors, that is 24 equivalent solutions in case $N_R = 4$. A way to overcome this is to use a formulation exploiting the independent sets of graph $\mathcal{G} = (\mathcal{B}, I)$, where an independent set is a set of vertices that are not linked by an edge. Basically, two beams in the same independent set can be allocated to the same reflector, so a way to compute a solution is to select a set of at most N_R independent sets in \mathcal{G} which can cover all the target polygons. Equivalently, a way to compute a solution is to select a set of N_R cliques of the complement graph \mathcal{G}' of \mathcal{G} Fig. 4 illustrates an independent set of beams associated with one of the industrial instances that we have.

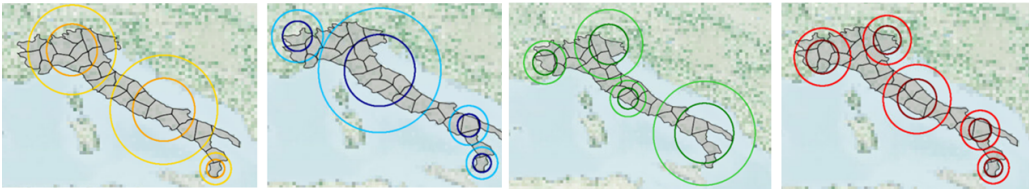


Figure 4: Column generation and independent sets: four independent sets on instance Italy, defining a solution (one color per independent set)

While enumerating all the independent sets allows us to break symmetries, it does not help handle the combinatorial explosion because the number of independent sets in \mathcal{G} can be huge for large instances. This leads us to propose a column generation approach where relevant independent sets are generated during the optimization process. This idea is inspired from works on the use of column generation for graph coloring [22]. In the following, we denote by \mathcal{K} the set of independent sets considered at a given search step. $a_{p,k}$ is a parameter equal to 1 if there exists a beam b in independent set $k \in \mathcal{K}$ such that b fully covers polygon p , and 0 otherwise.

6.1. Global description of the method

The basic idea behind column generation is to solve a master problem considering only on a small subset of variables (columns) at a given time, and to iteratively identify, through the resolution of a sub-problem, new variables (columns) that have the potential to improve the objective function. In our case, each column is a 0/1 variable corresponding to the selection of an independent set.

In the master problem, the selection of independent sets is driven by two objectives: first select at most N_R independent sets covering all the target

polygons (*i.e.*, find a feasible solution), and then select the independent sets involving the smallest beams (*i.e.*, improve the SRS criterion). Following these two objectives, the column generation approach proposed is composed of two phases, each phase having its own master problem.

In both phases, the sub-problem identifies whether there are any new columns that can improve the objective function. It is formulated as a maximum weighted independent set problem (MWIS) and uses the value of the dual variables of the master problem to compute the reduced cost of each potential column. If the reduced cost is negative, adding this column can potentially improve the solution. A global view of the column generation procedure is given in Fig. 5, and this procedure is detailed thereafter.

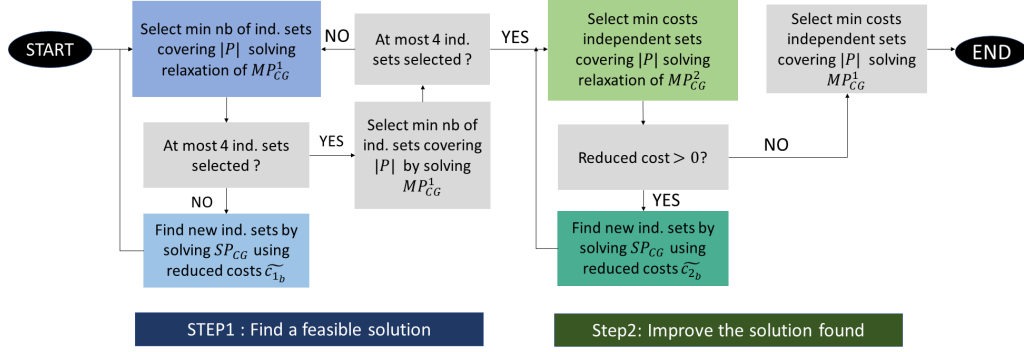


Figure 5: Column generation procedure

6.2. Phase 1: find a feasible solution

Master problem. The master problem MP_{CG}^1 used for the first phase is given in Equations (2a-2c). Its goal is to select independent sets which cover the polygons (decision variables $z_k \in \{0, 1\}$). Objective function (2a) minimizes the number of independent sets selected, or equivalently the number of reflectors required, while Constraints (2b) impose that all polygons must be covered by at least one independent set.

$$MP_{CG}^1 = \text{minimize} \quad \sum_{k \in \mathcal{K}} z_k \quad (2a)$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} a_{p,k} z_k \geq 1, \quad \forall p \in P, \quad (2b)$$

$$z_k \in \{0, 1\}, \quad \forall k \in \mathcal{K} \quad (2c)$$

Sub-problem. If the number of independent sets is greater than N_R in the solution found by the master problem, the sub-problem constructs a new independent set given the beam database, and the master problem is solved again. To define the sub-problem, let π_{1p} be the dual variables of Constraints (2b) for the polygons $p \in P$, and $\tilde{\pi}_{1p}$ be the values of these dual variables in the current solution. The dual constraint of variable z_k can be written

$$\sum_{p \in P} a_{p,k} \pi_{1p} \leq 1$$

and the reduced cost of a beam b is defined by

$$c_{1b}^{\tilde{}} = \sum_{p \in P_b} \tilde{\pi}_{1p} - 1$$

With u_b a variable representing the selection of a beam b in a new independent set, the sub-problem is written as:

$$SP_{CG} = \text{maximize} \quad \sum_{b \in B} c_{1b}^{\tilde{}} u_b \quad (3a)$$

$$\text{subject to} \quad u_{b_1} + u_{b_2} \leq 1, \quad \forall b_1, b_2 \in I, \quad (3b)$$

$$u_b \in \{0, 1\}, \quad \forall b \in \mathcal{B} \quad (3c)$$

The relaxations of MP_{CG}^1 and SP_{CG} are successively solved until the sub-problem cannot find any improving independent set or until the master problem finds a solution using at most N_R independent sets. In the first case, we have proven that the problem is infeasible with the set of beams in the database. In the second case, we solve the integer version of MP_{CG}^1 using integer programming. In our experiments, such a strategy always allows us to find a feasible solution. The study of branch-and-price methods integrating branch-and-bound and column generation is left for future work.

6.3. Phase 2: improve the solution found

Master problem. Once a feasible solution composed of at most N_R independent sets covering all polygons in P is found, we solve the master problem MP_{CG}^2 given in Equations (4a-4d) to optimize the selected beam sizes. Objective function (4a) consists in minimizing the SRS criteria, given that the cost C_k of an independent set $k \in \mathcal{K}$ is defined as $C_k = \sum_{b \in k} s_b^2$. In the first phase, the criteria of the master problem is transformed into Constraint (4b): the

number of independent sets selected must not exceed N_R . Last, Constraints (4c) ensure that each polygon is covered at least once by a beam.

$$MP_{CG}^2 = \text{minimize} \quad \sum_{k \in \mathcal{K}} C_k z_k \quad (4a)$$

$$\text{subject to} \quad \sum_{k \in \mathcal{K}} z_k \leq N_R \quad , \quad (4b)$$

$$\sum_{k \in \mathcal{K}} a_{p,k} z_k \geq 1, \quad \forall p \in P, \quad (4c)$$

$$z_k \in \{0, 1\}, \quad \forall k \in \mathcal{K} \quad (4d)$$

Sub-problem. Let α be the positive dual variable of Constraint (4b) and π_{2p} the dual variables of Constraints (4c) for the different polygons $p \in P$. The dual constraint of variable z_k can be written

$$\sum_{p \in P} a_{p,k} \pi_{2p} - \alpha \leq C_k$$

Hence, to find a violated dual constraint, one searches for a beam set k verifying

$$\sum_{p \in P} a_{p,k} \tilde{\pi}_{2p} - \tilde{\alpha} > C_k$$

where $\tilde{\pi}_{2p}$ and $\tilde{\alpha}$ are the values of the dual variables in the current solution of the master problem. Let \tilde{c}_{2_b} be the reduced cost of a beam b , equal to

$$\tilde{c}_{2_b} = \sum_{p \in P_b} \tilde{\pi}_{2p} - s_b^2 - \tilde{\alpha}$$

The sub-problem is expressed as in Equations (3a-3c) by replacing \tilde{c}_{1_b} by \tilde{c}_{2_b} .

6.4. Generation of independent sets

The role of the sub-problem is to search for maximum weighted independent sets in \mathcal{G} that have a negative reduced cost. To enumerate all maximum independent sets, the Bron-Kerbosh algorithm is often used [3]. Even if existing improvements can boost the original algorithm, such as degeneracy ordering, pivoting heuristics, or branch-and-bound [20], finding all the maximum cliques in a graph is NP-hard anyway. Initially, the master problem MP_{CG}^1 requires an initial set of independent sets \mathcal{K} . We first tried to consider only the beams in \mathcal{B}_{1P} and \mathcal{B}_{2P} that are respectively defined from 1

and 2 border polygons, preferred for their small size in the final solution. It is possible to enumerate all cliques for these sets of beams for instances with less than 50 polygons, using the Bron-Kerbosch algorithm, and adapted by Tomita, Tanaka and Takahashi [21]. We can also find initial independent sets heuristically, taking care that each polygon is covered by at least one independent set in initial set \mathcal{K} , based on a greedy clique expansion algorithm. We browse all polygons $p \in P$ and first select the node corresponding to the beam covering only the polygon p . The algorithm randomly selects at each step a vertex (a beam in our case), and then recursively adds neighbors of this vertex in the complement graph of \mathcal{G} as long as they form a clique. This heuristic method is always far better in terms of computational time, it never exceeds 6s instead of thousands of seconds for the exact method.

Second, instead of solving an ILP minimizing the sum of reduced cost \tilde{c}_{1_b} in the sub-problems, new independent sets can be produced heuristically. For this, we generate a clique with the greedy clique expansion algorithm, until finding a clique which doesn't belong to \mathcal{K} with a negative reduced cost. We used the heuristic generation of clique or not according to the instance, and the best solution was retained in the final table of results.

7. Benders Decomposition

The column generation approach presented before solves a master problem involving a subset of the decision variables and iteratively generates new relevant variables based on the sub-problem. An opposite approach consists in solving a master problem involving a subset of the constraints, and generating new relevant constraints at the level of the sub-problem. On this line, we study Benders decomposition methods. In classical Benders decomposition, the master problem works on a set of search variables, and the sub-problem solves a linear program over a set of remaining variables conditioned by the values found for the master problem's variables; The sub-problem evaluates whether the solution of the master problem is actually feasible or not and generates Benders cuts (new constraints) that prevent the master problem from making similar infeasible decisions over its search variables at future iterations. As the formalization of the beam-layout optimization problem introduced in Equations (1a-1e) does not involve continuous decision variables, we exploit Logic-Based Benders Decomposition (LBBD [14]).

7.1. Global description of the method

The main idea of the method proposed is to avoid expressing all the constraints related to the compatibility between beams allocated to the same reflector, because the number of these constraints is quadratic in the number of beams (Constraints 1d). At a given search step, the master problem considers only a subset of these compatibility constraints and proposes a solution minimizing the SRS criterion, while the sub-problem identifies conflicts between beams allocated to the same reflector and generates new cuts for the master problem. These cuts are iteratively added to the master problem, until a feasible solution is found. One key point in the method is that to get stronger cuts, the master problem and the sub-problem reason not only about the allocation of beams to reflectors (decision variables $x_{b,r} \in \{0, 1\}$ for $b \in \mathcal{B}$ and $r \in R$), but also about the allocation of polygons to reflectors (decision variables $y_{p,r} \in \{0, 1\}$ for $p \in P$ and $r \in R$). Here, we say that polygon p is allocated to a reflector r when we choose to cover p by a beam allocated to r , and as detailed thereafter, these decisions allow us to get stronger cuts mainly because the number of polygons is lower than the number of beams.

7.2. Master problem

The model is presented in Equations (5a-5g). It uses the decision variables $x_{b,r} \in \{0, 1\}$ and $y_{p,r} \in \{0, 1\}$ mentioned before. Objective function (5a) consists in minimizing the total beam cost. Constraints (5b) ensure that each polygon is associated with one reflector. Constraints (5c) ensure that every polygon is covered by at least one beam. Constraints (5d) express that each beam can be allocated to at most one reflector. Constraints (5e) can be illustrated with an example. Basically, if two polygons p_1, p_2 are allocated to the same reflector r and are very close to each other, it is not possible to use two independent beams over r to cover p_1 and p_2 respectively. This means that there must exist a beam b allocated to r that contains both polygons p_1 and p_2 . In other words constraint $y_{p_1,r} + y_{p_2,r} \leq 1 + \sum_{b \in \mathcal{B}_{\{p_1, p_2\}}} x_{b,r}$ must be satisfied. In this case, we say that $\{p_1, p_2\}$ is a *polygon cluster*, and the constraint added is called a *polygon grouping constraint* over cluster $\{p_1, p_2\}$. This example can be generalized to polygon clusters of sizes greater than 2, where a polygon cluster $h = \{p_1, \dots, p_n\}$ is a set such that if all polygons in h are covered by the same reflector r , then there must exist a common (potentially large) beam allocated to r that covers

all the polygons in h . This generalization leads to the polygon grouping constraints expressed in Constraints (5e), that exploit as an input a set of polygon clusters \mathcal{H} . Note that these constraints can also be used to express that a set of polygons cannot be covered by the same reflector when this creates a beam of radius greater than s_{MAX} (case $\mathcal{B}_h = \emptyset$ in Constraints (5e)). In the initial master problem, set \mathcal{H} contains all polygons clusters $\{p_1, p_2\}$ composed of two polygons such that beams $W(\{p_1\})$ and $W(\{p_2\})$ have an insufficient separation distance owing to parameter κ . As shown later, the sub-problem is used to generate new relevant polygon clusters, giving the master problem a better view of the actual beam activations required given the reflector associated with each polygon.

$$MP_B = \min \quad \sum_{b \in \mathcal{B}} x_{b,r} \times s_b^2 \quad (5a)$$

$$\text{s.t.} \quad \sum_{r \in R} y_{p,r} = 1, \quad \forall p \in \mathcal{P}, \quad (5b)$$

$$\sum_{b \in \mathcal{B}_p, r \in R} x_{b,r} \geq 1, \quad \forall p \in \mathcal{P}, \quad (5c)$$

$$\sum_{r \in R} x_{b,r} \leq 1, \quad \forall b \in \mathcal{B}, \quad (5d)$$

$$\sum_{p \in h} y_{p,r} \leq |h| - 1 + \sum_{b \in \mathcal{B}_h} x_{b,r}, \quad \forall h \in \mathcal{H}, \forall r \in R, \quad (5e)$$

$$x_{b,r} \in \{0, 1\} \quad \forall b \in \mathcal{B}, \forall r \in R, \quad (5f)$$

$$y_{p,r} \in \{0, 1\} \quad \forall p \in \mathcal{P}, \forall r \in R \quad (5g)$$

7.3. Sub-problem

A solution of the master problem selects a set of beams $\tilde{\mathcal{B}}$, allocates a reflector to each selected beam, and allocates a reflector to each polygon. The master problem tends to propose a solution involving small beams given its objective function, but two beams using the same reflector in this solution may actually be in conflict because the master problem is not aware of all the beams incompatibility pairs in I . In the sub-problem, we identify these conflicts and generate new cuts for the master problem, that is new polygon clusters and new polygon grouping constraints. For instance, in Fig. 6a, we consider two beams b_1 and b_2 covering polygons $\{p_7, p_9, p_{10}\}$ and $\{p_5\}$ respectively. These beams are associated with the same reflector in the master problem but they are too close to each other. The sub-problem then computes a new set of polygons $h = \{p_5, p_7, p_9, p_{10}\}$ to add to the set of polygon

clusters \mathcal{H} . From this point forward, in the master problem, the polygons in cluster h cannot be allocated to the same reflector without activating one of the beams covering all these polygons together, for example the beam in Fig. 6b. The sub-problem must also compute $\mathcal{B}_h = \{b \in \mathcal{B} \mid h \subseteq P_b\}$, and the new polygon grouping constraints associated with cluster h are

$$\forall r \in \mathcal{R}, \sum_{p \in h} y_{p,r} \leq |h| - 1 + \sum_{b \in \mathcal{B}_h} x_{b,r}$$

Note that from a theoretical point of view, these cuts are not necessarily valid. Indeed, even if two beams $W(P_1)$ and $W(P_2)$ obtained from two subsets of polygons P_1, P_2 are insufficiently separated according to parameter κ , there may exist another polygon p such that $W(P_1 \cup \{p\})$ and $W(P_2)$ are not in conflict thanks to the change of the radius of curvature of the first beam following the addition of p (see Appendix B for an example). For the sake of convergence speed, we ignore such non-monotonous effects, while keeping in mind that the approach proposed then does not necessarily produce the optimal solution even if it converges.

7.4. Cut reduction

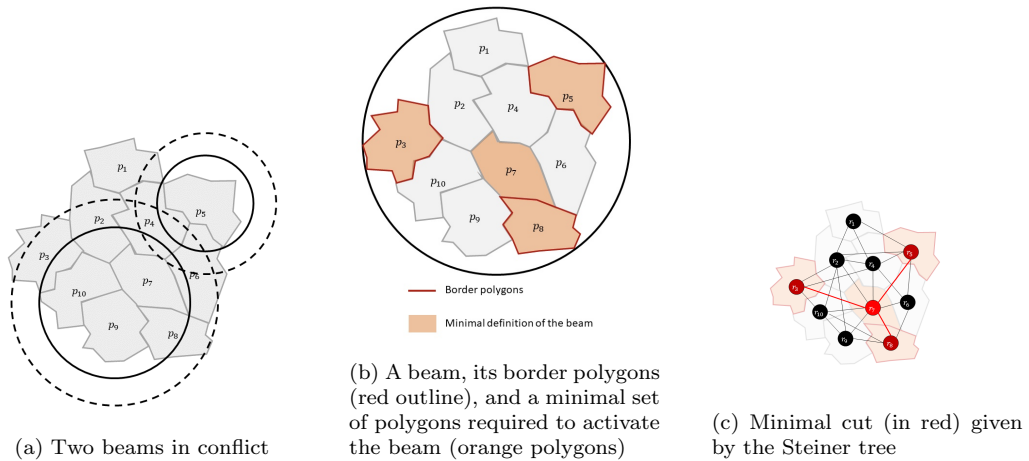


Figure 6: Minimum definition of a beam

As the process progresses, the size of the cuts tends to increase: the master problem selects larger and larger beams to avoid conflicts, and the sizes of the polygon clusters h added to \mathcal{H} grows. However, the efficiency of

our method depends on the strength of the cuts generated because tighter constraints help the master problem make better decisions in subsequent iterations and reduce the number of iterations required to converge. Therefore, we introduce techniques to produce relevant polygon clusters of smaller sizes. The approach is illustrated in Fig. 6b. A beam covers polygon cluster $h = \{p_1, \dots, p_{10}\}$, but only a few of them need to be allocated to the same reflector to force the activation of the beam. For example, if we remove polygons p_1, p_2, p_4, p_6, p_9 , and p_{10} , and if we compute beam $W(h')$ associated with polygon cluster $h' = \{p_3, p_5, p_7, p_8\}$, we obtain the exact same beam as $W(h)$. Hence, allocating the same reflector to the polygons in $\{p_3, p_5, p_7, p_8\}$ already forces the activation of a beam as large as the black one, even if p_1, p_2, p_4, p_6, p_9 and p_{10} use a different reflector.

To compute a smaller polygon cluster sufficient to activate beam $b = W(h)$, we first identify the border polygons of b that have a point placed on the smallest covering circle of P_b . If these polygons are too far apart to activate the beam, we add coupling polygons between them.

As depicted in Fig. 6c, (1) we construct the elementary beams $W(p)$ covering the polygons $p \in P_b$; (2) we build the graph Γ containing one vertex per elementary beam $W(p)$ and one edge between vertices associated with two beams that are too close to each other to be placed on the same reflector; (3) we compute a subset of edges of minimum size such that the sub-graph induced by these edges is connected and contains all the vertices associated with the elementary beams of the border polygons. The last problem is a Steiner tree problem, which we solve using a fast approximation algorithm proposed by Melhorn [17]. After that, we get a potentially reduced polygon cluster $h' \subseteq h$, and if $h' \neq h$ we add both h and h' to the master problem.

7.5. Growing pool of beams

To increase the performance of the Benders method over the largest instances containing more than 100 polygons, we introduce a last (optional) mechanism inspired from our previous matheuristic method [16]. Initially, we consider a pool of beams is formed by all the beams defined from 1 and 2 border polygon(s). If a conflict between two beams b_1 and b_2 is detected in the sub-problem, we add the cuts associated with this conflict as exposed in Section 7.3, and we also add a new beam $b_3 = W(P_{b_1} \cup P_{b_2})$ to the beam database. Following this addition, we also update sets \mathcal{B}_h for all polygon clusters $h \in \mathcal{H}$ if needed. As shown in the experimental results, this extension allows us to find solutions for our largest instance.

8. Experimental results

We have implemented the methods presented in section 6 and 7 in python3.8. ILP models : (1, . . . , 5) were solved with CPLEX12.10. The runs were made on a server with 96 cores of an Intel(R) Xeon(R) Gold 5318Y CPU @2.10GHz processor and 62GB of RAM.

8.1. Instances

We built a set of realistic instances presented in Table 3. The complexity in the resolution of a given instance can first be evaluated by examining its number of polygons referred as "Nb polygons" in the table. The bigger are the polygons, the more difficult it will be to group them in beams without exceeding the maximum beam size. The inter-connectivity between the polygons is important too: the closer they are from each other, the more difficult it will be to find beams of small sizes with feeds that do not overlap. In order to have a first indication of the difficulty of an instance, we have constructed the set of beams \mathcal{B}_{1P} where each beam cover one unique polygon. We can then construct the graph $\mathcal{G}_{1P} = (\mathcal{B}_{1P}, E_{1P})$, and evaluate its number of edges $|E_{1P}|$ and its chromatic number γ_{1P} . The higher is the number of edges in \mathcal{G}_{1P} , the more difficult it is to find a set of beams that can be assigned to the same reflector. Similarly, the higher the chromatic number of \mathcal{G}_{1P} , the further we are from a feasible solution that requires a chromatic number lower than or equal to N_R . These assumptions can be directly verified in the table of results; for example, on the Italy instance, the number of polygons is higher than on the France instance, but the resolution time is significantly lower. Similarly, the lower is the maximum radius of \mathcal{B}_{1P} , the fewer beam are allowed, which tends to reduce the combinatorial complexity of the problem. *If the paper is accepted, the instances will be available on a github* We have grouped different beam-layout solutions plot on several instance in [Appendix A](#).

8.2. Table of results

The results and comparison of the methods are gathered in Table 5. " T_M " and " T_S " respectively refer to the sum of total computational time of the master problem and the sub-problem on each iteration, while " T " is the total computational time. The number of iterations is referred in column "It". For the solutions obtained with the column generation approach, it is the sum of the number of iteration of step1(finding a feasible solution) and

Instance	Nb polygons	$ E_{1P} $	γ_{1P}	Max radius
France	22	144	10	0.2192
GuySur	23	121	8	0.2114
Italy	30	126	8	0.1263
Venezuela	34	219	9	0.2903
Morocco	39	180	8	0.1743
UK	40	247	10	0.1494
Spain	44	337	11	0.1827
EU51	51	402	11	0.1797
EU52	52	294	10	0.2192
BlackSea	50	333	10	0.2018
WestEurope	92	467	9	0.1554
Egypte	90	695	11	0.1579
EU102	102	893	12	0.2192
EU136	136	969	11	0.2192

Table 3: Presentation of the instances

step2 (improving the solution), we don't differentiate the two steps because the number of iterations of step 1 is always 1. With the column generation method, we differentiate when the sub-problem is solve using an heuristic algorithm (CG_H) and when an exact method is used in the sub-problem (CG_H), as explained in section 6.4. In column "GapBKS", the SRS Gap relatively to the Best Know Solution is given, which is the solution given by the ILP on the instances form France to EU52 in the table's order, and the solution given by LBBd method on instances EU52 to EU136. As the column generation approach and the heuristic needed a stop criteria, we gave the same time limit to both of these methods : 180s, which represent the maximum computational time of the heuristic on the instance where all methods could find a solution (which each 180s). Thanks to the beam database construction method presented in section 3.3, the ILP was solvable in reasonable time on the smallest instances. The LBBd is definitely more efficient in terms of computational time and quality of the solutions found on all instances, it beats the heuristic, the column generation approach and the ILP method computationally, and find the best solution on all instances.

Optimality. On the instances were we could manage to find an optimal solution with the model 1 of sub-section 5, LBBd was always able to find the optimal solution in shorter time than the ILP, which represents 10 instances over 13. The Column generation approach can find the optimal solution on

Instance	Method	T (s)	T_M (s)	T_S (s)	It	$ \tilde{\mathcal{B}} $	$MaxS$	MSRS	SRS	GapBKS (%)
France	<i>Heur.</i>	180	60.5	71	16	7	0.344	0.0672	0.470	1.7
	<i>ILP</i>	7.9			1	9	0.372	0.0514	0.462	0
	<i>CG_E</i>	180	4.6	165.5	124	6	0.376	0.0785	0.471	1.85
	<i>CG_H</i>	180	0.5	178.7	2	11	0.399	0.042	0.497	7.42
	<i>LBBD</i>	3.7	3.6	3.4	4	9	0.372	0.0514	0.462	0
GuySur	<i>Heur.</i>	180	107.5	2.9	21	6	0.479	0.0886	0.532	0
	<i>ILP</i>	9.9			1	6	0.478	0.0886	0.532	0
	<i>CG_E</i>	180	3.5	174.9	80	6	0.479	0.0886	0.532	0
	<i>CG_H</i>	180	0.7	178.4	4	8	0.38	0.0739	0.591	11.16
	<i>LBBD</i>	1.1	0.7	0.5	1	6	0.479	0.0886	0.532	0
Italy	<i>Heur.</i>	180	113.1	5.9	10	13	0.229	0.0153	0.198	0.7
	<i>ILP</i>	8.1			1	12	0.194	0.0164	0.197	0
	<i>CG_E</i>	180	4.6	165.7	100	10	0.229	0.0213	0.213	7.8
	<i>CG_H</i>	180	0.8	179.1	7	8	0.27	0.0288	0.231	16.97
	<i>LBBD</i>	0.7	3.6	3.2	3	12	0.194	0.0164	0.197	0
Venez.	<i>Heur.</i>	180	120.5	3.7	7	8	0.589	0.148	1.182	0.25
	<i>ILP</i>	51.4			1	7	0.588	0.169	1.179	0
	<i>CG_E</i>	180.3	1.6	174.5	22	9	0.588	0.146	1.312	4.6
	<i>CG_H</i>	180	1.1	178.9	5	6	0.596	0.205	1.23	4.33
	<i>LBBD</i>	48.8	32.1	16.17	11	7	0.588	0.169	1.179	0
Morocco	<i>Heur.</i>	180	102.9	9.2	8	11	0.374	0.0518	0.570	0.18
	<i>ILP</i>	32			1	13	0.374	0.0438	0.569	0
	<i>CG_E</i>	180	4.1	166.5	36	12	0.398	0.0533	0.634	12.39
	<i>CG_H</i>	180	0.9	178.9	3	9	0.434	0.102	0.713	25.29
	<i>LBBD</i>	7.6	4.23	3.35	5	13	0.377	0.0438	0.569	0
UK	<i>Heur.</i>	180	119.4	4.3	5	9	0.226	0.0229	0.205	0
	<i>ILP</i>	195.7			1	12	0.229	0.0171	0.205	0
	<i>CG_E</i>	180	1.4	172.1	6	12	0.259	0.0258	0.232	13.36
	<i>CG_H</i>	180	1.1	178.2	5	9	0.229	0.0234	0.211	2.97
	<i>LBBD</i>	98.5	57.9	47.56	5	12	0.229	0.0171	0.205	0
Spain	<i>Heur.</i>	180	132.07	12.5	3	8	0.389	0.0528	0.4223	4.7
	<i>ILP</i>	539.2			1	9	0.399	0.0452	0.407	0
	<i>CG_E</i>	180	1.4	172.1	6	12	0.371	0.058	0.465	14.29
	<i>CG_H</i>	180	1.1	178.2	5	9	0.359	0.0563	0.45	10.7
	<i>LBBD</i>	36.5	20.2	16.31	5	9	0.399	0.0452	0.407	0
BlackSea	<i>Heur.</i>	180	137.7	12.5	3	8	0.477	0.0786	0.55	1.9
	<i>ILP</i>	601.8			1	7	0.484	0.0771	0.540	0
	<i>CG_E</i>	180	2.1	172	2	12	0.418	0.0523	0.628	16.5
	<i>CG_H</i>	180	1.9	176.1	4	4	0.489	0.155	0.62	14.9
	<i>LBBD</i>	73.9	38.26	35.65	3	7	0.484	0.0771	0.540	0
EU51	<i>Heur.</i>	180	140.7	8.1	2	10	0.424	0.0642	0.642	12.4
	<i>ILP</i>	1371.7			1	17	0.463	0.0815	0.571	0
	<i>CG_E</i>	180	2.7	172.5	1	8	0.489	0.809	0.647	3.7
	<i>CG_H</i>	180	2.1	171.6	2	8	0.489	0.809	0.6478	3.7
	<i>LBBD</i>	25.8	11.9	13.91	2	7	0.463	0.0815	0.571	0

Table 4: All results, first table

EU52	<i>Heur.</i>	180	79.9	47.7	4	17	0.474	0.0367	0.624	0
	<i>ILP</i>	175.18			1	17	0.474	0.0367	0.624	0
	<i>CG_E</i>	180	2.5	172.5	6	12	0.4777	0.041	0.830	45.5
	<i>CG_H</i>	180	2.5	169.58	2	11	0.485	0.0756	0.829	45.2
	<i>LBBD</i>	26.02	13.28	12.74	6	17	0.474	0.0367	0.624	0
WestEU	<i>Heur.</i>	180	125.19	29.92	2	33	0.33	0.0283	0.934	27.96
	<i>LBBD</i>	192.73	93.07	99.66	6	41	0.209	0.0178	0.73	0
Egypt	<i>Heur.</i>	739.4	750.9	36.96	3	12	0.481	0.0925	1.11	24.9
	<i>LBBD</i>	674.72	320.23	354.49	7	15	0.493	0.0593	0.889	0
EU102	<i>Heur.</i>	560.4	500.89	41.75	2	11	0.6987	0.118	1.2976	16.71
	<i>LBBD</i>	9564.37	4212.99	5351.38	12	19	0.619	0.0585	1.112	0
EU136	<i>Heur.</i>	445.3	371.83	48.6	2	21	0.5451	0.0588	1.2342	5.1
	<i>LBBD</i>	1693.88	940.57	753.31	11	31	0.533	0.0379	1.174	0

Table 5: All results, second table

one instance, but was not able to beat the heuristic.

Computational time. The minimum, maximum and mean speed-up (SpU) computational time for each method on all the instances where the four methods could find a solution in table 6, showing how fast is the LBBD compared to the other methods. The speed-up time is defined as the relative gap between the computational time of the compared method (t_m) to the computational time of LBBD (t_{LBBD}): $\frac{(t_m - t_{LBBD})}{t_m}$.

	Min. SpU (s)	Max. SpU (s)	Mean SpU (s)
Heuristic	-223.8	88.9	-0.1
ILP	31.3	98.1	149
CG	45.3	99.6	13.42

Table 6: Speed-up of LBBD method compared to Heuristic, ILP and CG

8.2.1. Impact of the settings of the LBBD decomposition method

Instance	Method	T	It	$ \tilde{\mathcal{B}} $	<i>MaxS</i>	MSRS	SRS
EU	Heur.	436	6	15	0.6870	0.08568	1.2501
EU	LBBD	649	21	20	0.56123	0.06211	1.2422

Table 7: Results on Europe with extension

Impact of the growing pool of beams. LBBD method wasn't performant on one of the biggest instance with 102 polygons. The initial beam database

was too huge, and the number of cuts necessary to obtain a feasible solution takes too much time to generate. Table 7 compare the results of 7.5 with the best heuristic result found. It is important to notice that this extension wasn't an improvement on the other smaller instances.

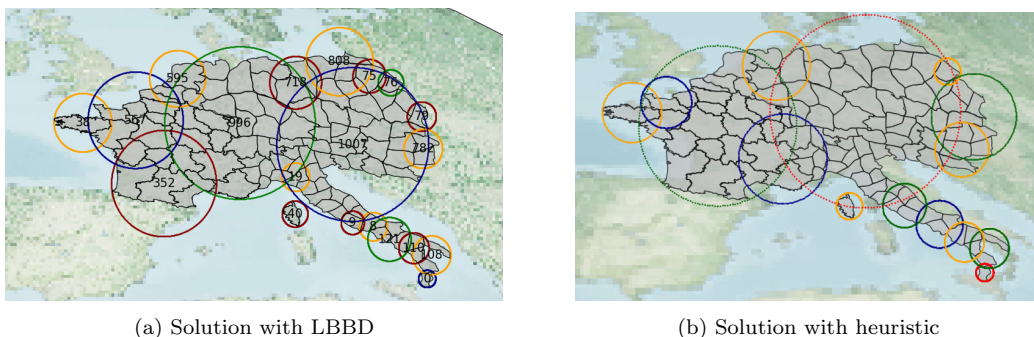


Figure 7: Solutions on 102 polygons Europe instance

Impact of the cut reduction technique. Figures 8 show on two instances how the cut reduction technique can decrease the computational time.

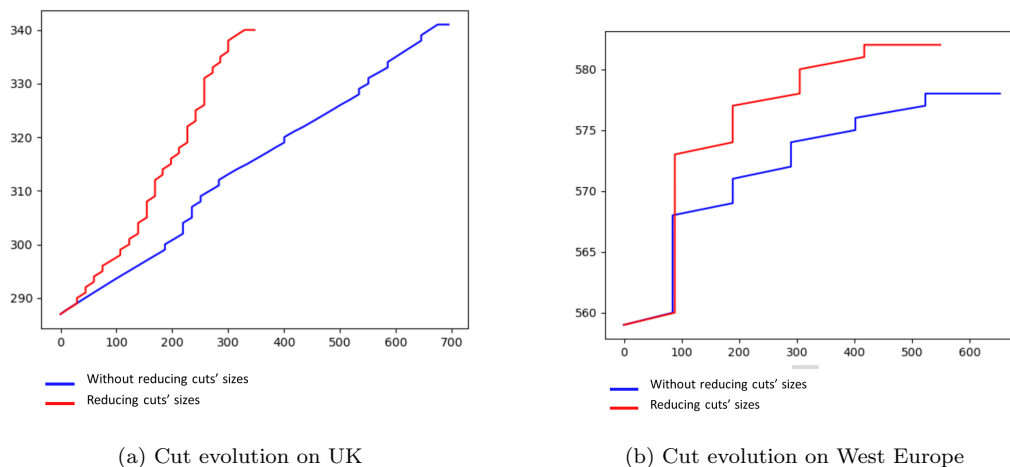
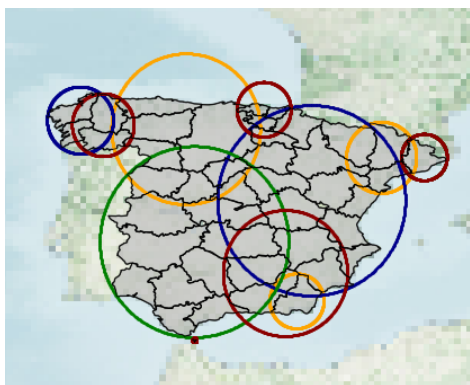


Figure 8: Number of cut evolution according to computing time for three different configuration, computational time on x-axis, number of cuts on y-axis

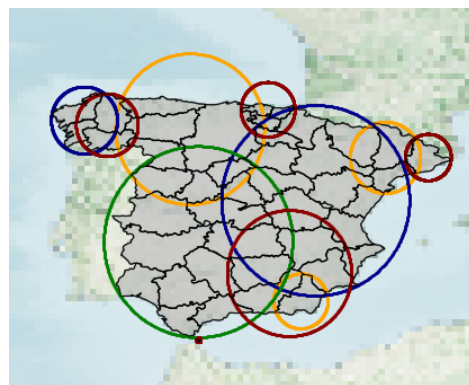
9. Conclusion

We propose to detail two decomposition methods to solve the beam-layout optimization problem on multi-beam broadcasting missions. The area of interest is divided in a set of polygons who needs to be covered entirely by at least one beam. In addition, considering the SFPB antenna technology, we propose to allocated the beams to the reflectors with graph coloring. The polygons need to be grouped in beams and the beams dispatched on the different reflectors available, such that the arrangement of their associated antenna elementary equipment call feeds is mechanically feasible, knowing that the smaller are the beams, the better will be the radio frequency performances. Combining graph coloring constraints and high combinatorial issues can give near-optimal solutions, leading us to decomposition methods. Results are compared with a heuristic and an exact ILP on realistic instances. The first method, column generation, can addresses problem symmetries and leaves space for future improvements. The second method, inspired by Logic-Based Benders Decomposition, outperforms all other methods, and was proven optimal for most of the instances. However, scaling up instances is challenging for both methods, but some improvements can allow us to provide solutions too. For the column generation, we can compute clique heuristically in the sub-problems instead of solving an integer linear program that might becomes unsolvable on the biggest instance. We tried to use a greedy algorithm but some improvements can be done to generate potentially good cliques. Concerning Benders decomposition, the biggest instance can be solved by an iterative process for generating beams in the sub-problem, but further analyses must be done on this line of research.

Appendix A. Solutions beam-layout plots

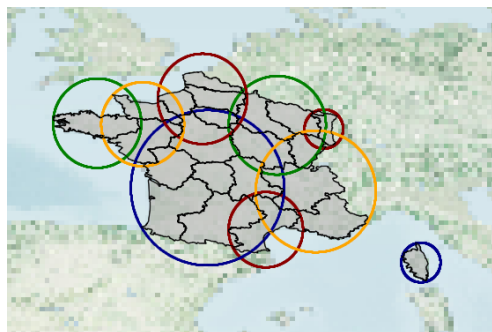


(a) LBBD solution on Spain

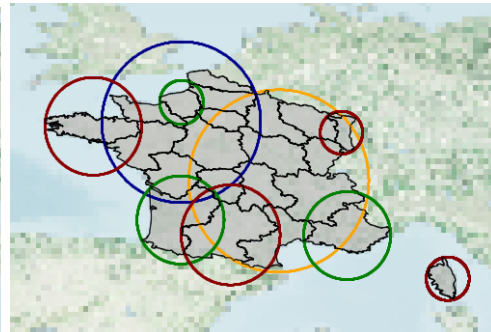


(b) Column Generation solution on Spain

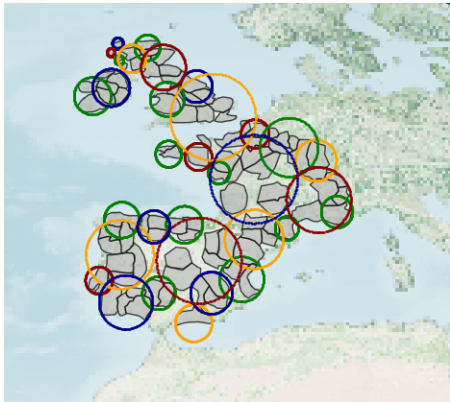
Figure A.9: Solutions plots, small instances



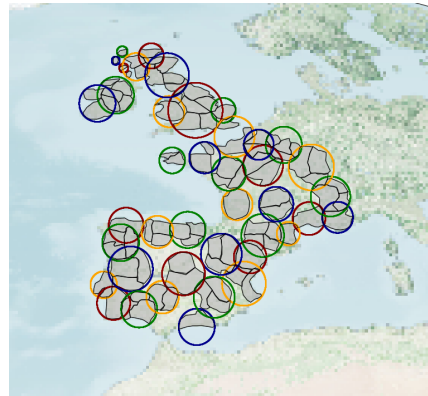
(a) LBBD solution on France



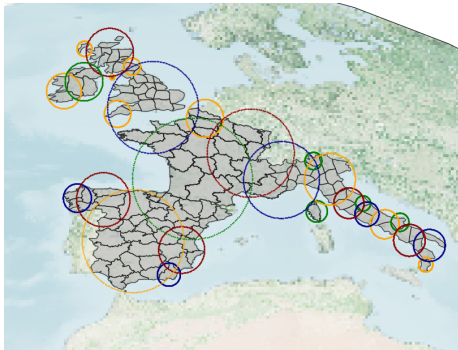
(b) Column Generation solution on France



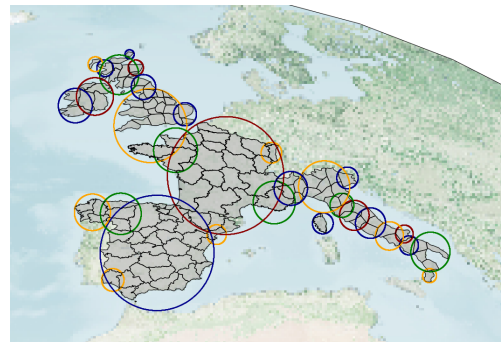
(a) Heuristic solution on West Europe



(b) LBBD solution on West Europe



(c) Heuristic solution on EU136



(d) LBBD solution on EU136

Figure A.11: Solutions plots, large instances

Appendix B. Explanation Benders is not exact

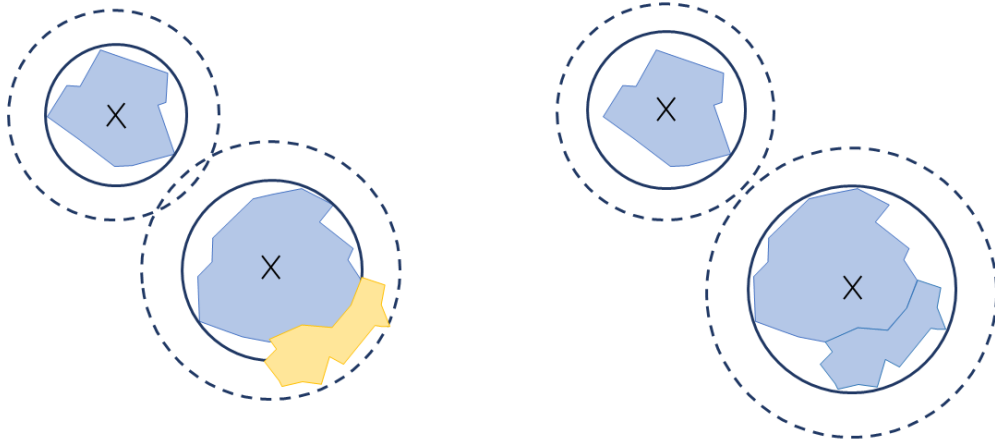


Figure B.12: (right) 2 polygons in separate beams associated to the same reflector are in conflict (left) The decentering of one of the beam by adding a third polygon can avoid the conflict

References

- [1] Angeletti, P., & Cubillos, J. L. (2021). Traffic balancing multibeam antennas for communication satellites. In *IEEE Transactions on Antennas and Propagation*, 69(12), 8291-8303.
- [2] Bánhelyi, B., Palatinus, E., & Lévai, B. L. (2015). Optimal circle covering problems and their applications. *Central European Journal of Operations Research*, 23(4), 815-832.
- [3] Bron, C., & Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9), 575-577.
- [4] Camino, J. T., Mourgues, S., Artigues, C., & Houssin, L. (2014, September). A greedy approach combined with graph coloring for non-uniform beam layouts under antenna constraints in multibeam satellite systems. In *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)* (pp. 374-381). IEEE.

- [5] Camino, J. T., Artigues, C., Houssin, L., & Mourgues, S. (2016, April). Mixed-integer linear programming for multibeam satellite systems design: Application to the beam layout optimization. In *2016 Annual IEEE Systems Conference (SysCon)* (pp. 1-6). IEEE.
- [6] Clark, B. N., Colbourn, C. J., & Johnson, D. S. (1990). Unit disk graphs. *Discrete mathematics*, 86(1-3), 165-177.
- [7] Das, G. K., Fraser, R., Lóopez-Ortiz, A., & Nickerson, B. G. (2012). On the discrete unit disk cover problem. *International Journal of Computational Geometry & Applications*, 22(05), 407-419.
- [8] Fowler, R. J., Paterson, M. S., & Tanimoto, S. L. (1981). Optimal packing and covering in the plane are NP-complete. *Information Processing Letters*, 12(3), 133-137.
- [9] Gaudry, A., Li, R., & Mak-Hau, V. (2023). FlexBeamOpt: hybrid solution methodologies for high-throughput GEO satellite beam laydown and resource allocation. *International Journal of Satellite Communications and Networking*, 41(6), 515-538. doi:10.1002/sat.1481
- [10] Li, R., Gaudry, A., & Mak-Hau, V. (2024). CG-FlexBeamOpt: Advanced solution methodology for high throughput GEO satellite beam laydown and resource allocation. *International Journal of Satellite Communications and Networking*.
- [11] Gräf, A., Stumpf, M., & Weißenfels, G. (1998). On coloring unit disk graphs. *Algorithmica*, 20, 277-293.
- [12] Hale, W. K. (1980). Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68(12), 1497-1514.
- [13] Hammill, C. W., & Dishaw, K. O. (2001). U.S. Patent No. 6,173,178. Washington, DC: U.S. Patent and Trademark Office.
- [14] Hooker, J. N., & Ottosson, G. (2003). Logic-based Benders decomposition. *Mathematical Programming*, 96(1), 33-60.
- [15] Kyrgiazos, A., Evans, B., & Thompson, P. (2013, October). Irregular beam sizes and non-uniform bandwidth allocation in HTS multibeam satellite systems. In *31st AIAA International Communications Satellite Systems Conference (ICSSC)*

- [16] Lescuyer, C., Artigues, C., Camino, J. T., & Pralet, C. (2024, February). Heuristic Methods for the Antenna-Constrained Beam Layout Optimization on Multibeam Broadcasting Mission. In *13th International Conference on Operations Research and Enterprise Systems*
- [17] Mehlhorn, K. (1988). A faster approximation algorithm for the Steiner problem in graphs. *Information Processing Letters*, 27(3), 125-128.
- [18] Rao, S. K. (2003). Parametric design and analysis of multiple-beam reflector antennas for satellite communications. *IEEE Antennas and Propagation Magazine*, 45(4), 26-34.
- [19] Reyunuru, M. S., Jethlia, K., & Basappa, M. (2021). The k-colorable unit disk cover problem. In *Computational Science and Its Applications–ICCSA 2021: 21st International Conference, 2021, Proceedings, Part I 21* (pp. 402-417).
- [20] San Segundo, P., Rodríguez-Losada, D., & Jiménez, A. (2011). An exact bit-parallel algorithm for the maximum clique problem. *Computers & Operations Research*, 38(2), 571-581.
- [21] Tomita, E., Tanaka, A., & Takahashi, H. (2006). The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical computer science*, 363(1), 28-42.
- [22] Mehrotra, A., & Trick, M. A. (1996). A column generation approach for graph coloring. *informatics Journal on Computing*, 8(4), 344-354.
- [23] Welzl, E. (2005, June). Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science, 1991 Proceedings* (pp. 359-370)